MIAOW: An Open Source GPGPU

www.miaowgpu.org

Vinay Gangadhar, Raghu Balasubramanian, Mario Drumond, Ziliang Guo,
Jai Menon, Cherin Joseph, Robin Prakash, Sharath Prasad, Pradip Vallathol,
Karu Sankaralingam

Vertical Research Group

University of Wisconsin - Madison

Executive Summary

- MIAOW is a **credible GPGPU implementation**
  - Compatible with AMD Southern Islands ISA
  - Runs OpenCL programs and prototyped on FPGA
  - Similar design to industry state-of-art
  - Similar performance to industry state-of-art*
  - Flexible and Extendable
- MIOAW's hardware design is **Open Source**
- Contributes to changing hardware landscape

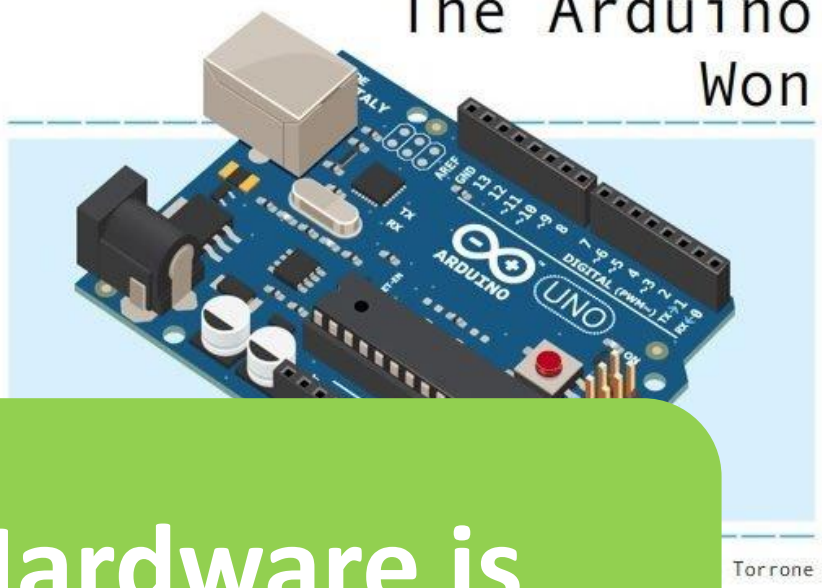
* *Frequency, Physical Design, Area goals relaxed*


CHANGE
AHEAD


THE INTERNET
Applications that drive computing are changing
Need innovative new hardware

131 maker fairs
750K people

Why
The Arduino
Won

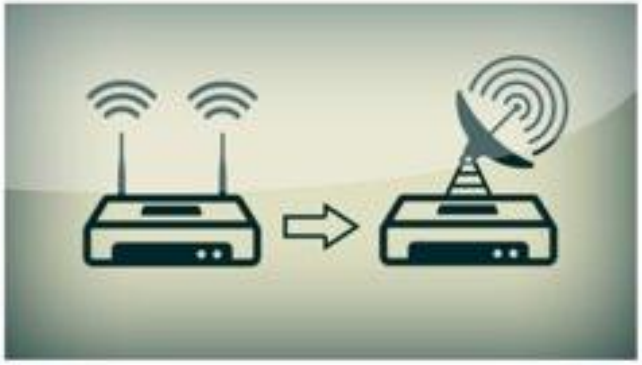


Torrone

Open Source **Hardware is** gaining momentum

lifehacker

Turn your $6



View more photos and videos

OPEN
Compute Project

Some Open Source Hardware Microprocessors

ZERO Open Source GPUs

Lessons from Open Source S/W

PHP, Linux, ruby, mysql, sqlite, apache, gcc
late 80s, early 90s

$0

Facebook, Twitter, Whatsapp, Instagram
Web 2.0

>$10 bill.

MIAOW, OpenCores, RISC-V etc..

$0

?

MIAOW Technical Overview

Demonstrate MIAOW is credible GPGPU

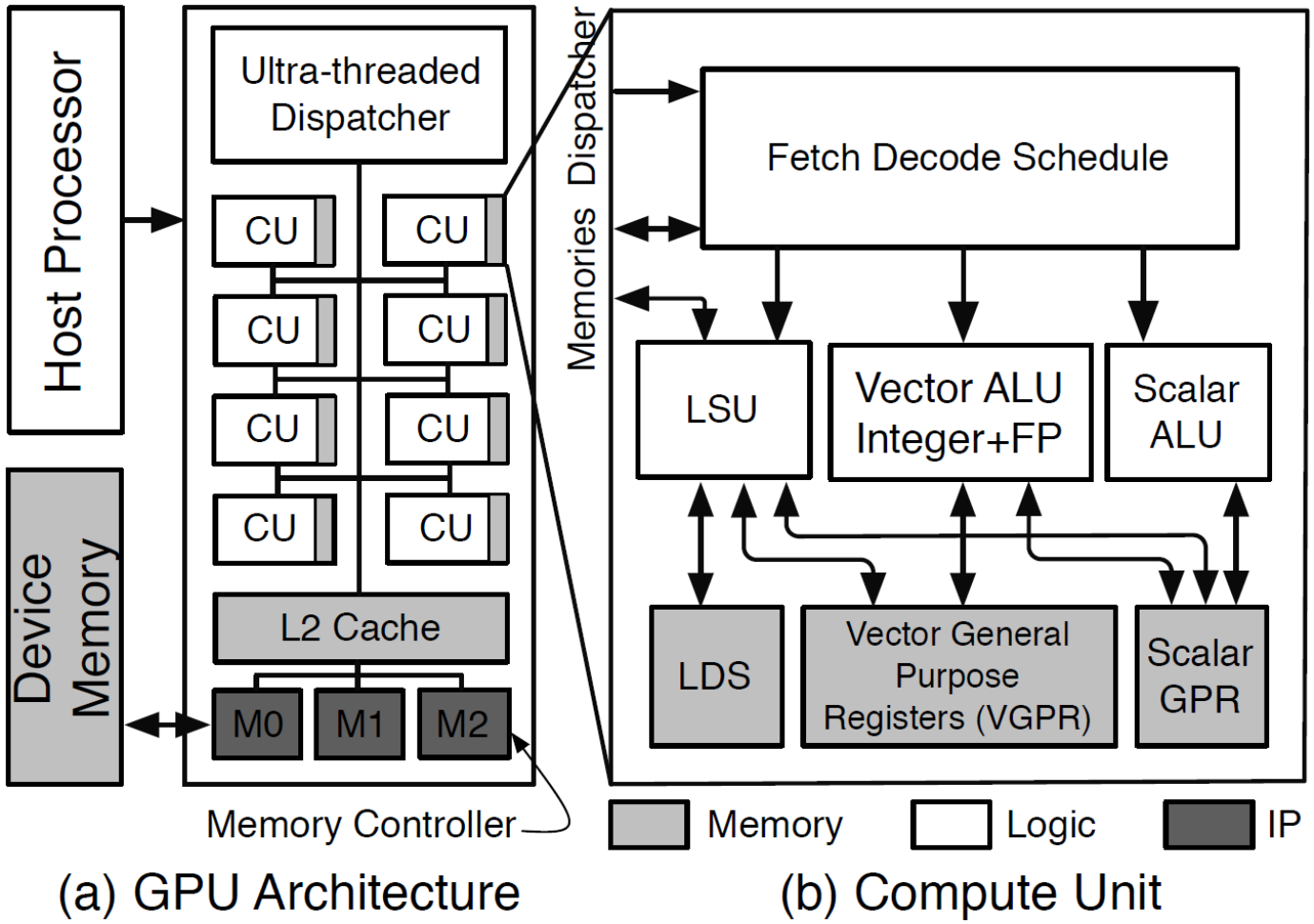
Implications and Possibility of Open Source Hardware

ISA Summary

Type		Instructions
Vector	ALU:	add, addc, sub, mad, madmk, mac, mul, max, max3, min, subrev
	Bitwise:	and, or, xor, not, mov, lshrrev, lshlrev, ashlrev, ashrrev, bfe, bfi, cndmask
	Compare:	cmp_{ lt, eq, le, gt, lg, ge, ne, ngt, neq }
Scalar	ALU:	add, addk, sub, max, min, mul, mulk
	Bitwise:	and, andn2, or, xor, not, mov, movk, lshl, lshr, ashr, saveexec
	Compare:	cmp_{ eq, lt, gt, ge, lt, le, eq, lg, gt, ge, lt, le }
	Conditional:	barrier, branch, cbranch, endpgm, waitcnt
Memory	Scalar_Mem:	load, buffer_load
	Vector_Mem:	tbuffer_load, tbuffer_store
	Date Share (LDS, GDS):	ds_read, ds_write

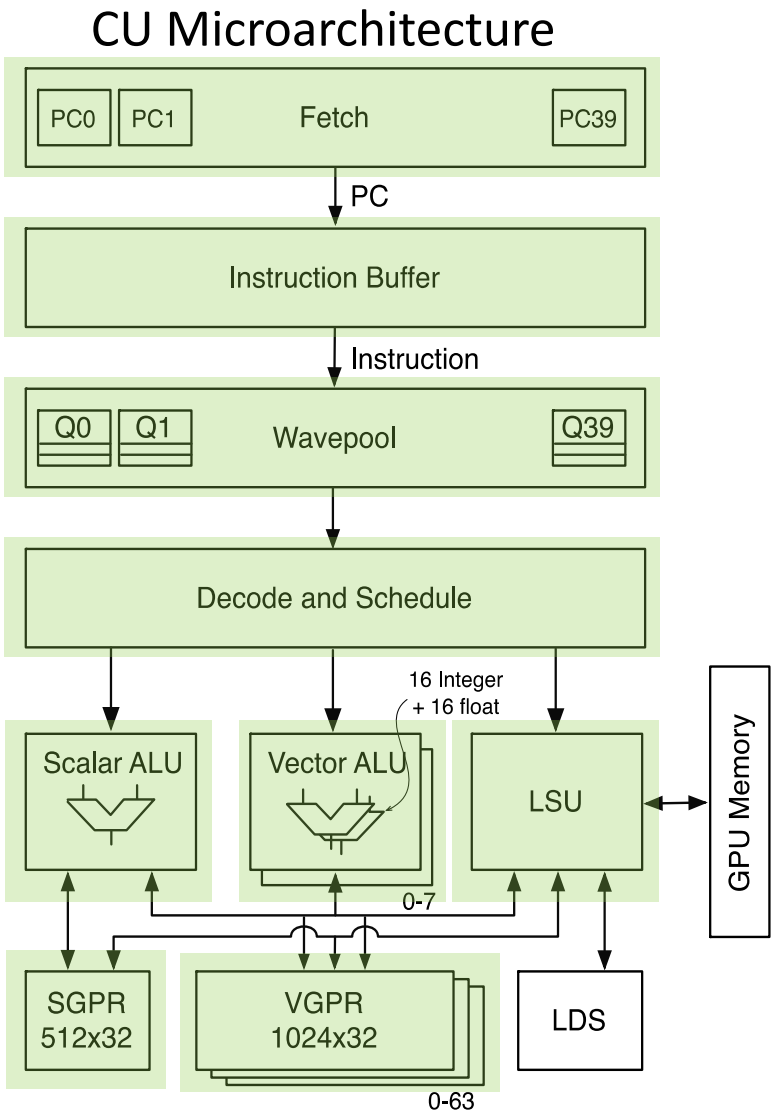
- 95 instructions
- Single-precision support only
- No graphics support (yet)

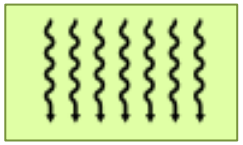
MIAOW Overview



MIAOW has 32 Compute Units (CUs)

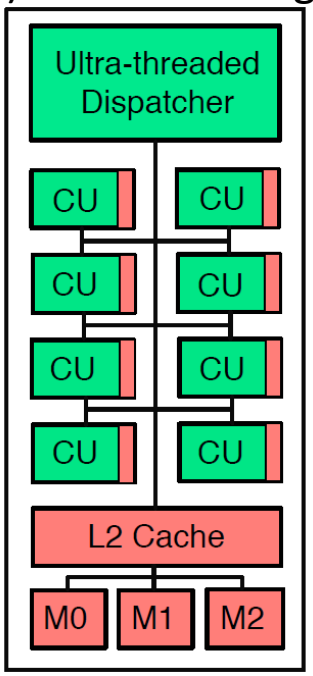
Hardware Organization



- Single Issue
- 40  Wavefronts
- 16-wide vector ALUs
- LSU – Memory operations

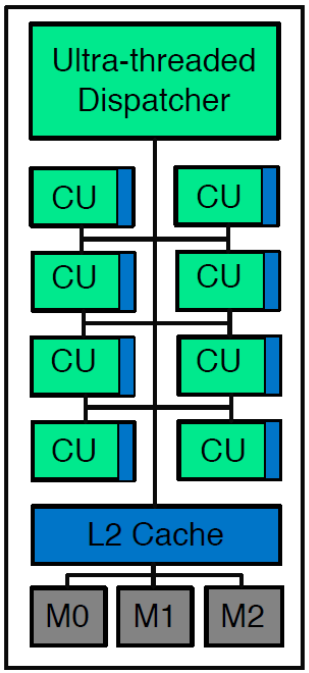
MIAOW Implementations

(a) Full ASIC Design



Area, Power from Floorplan and simulation

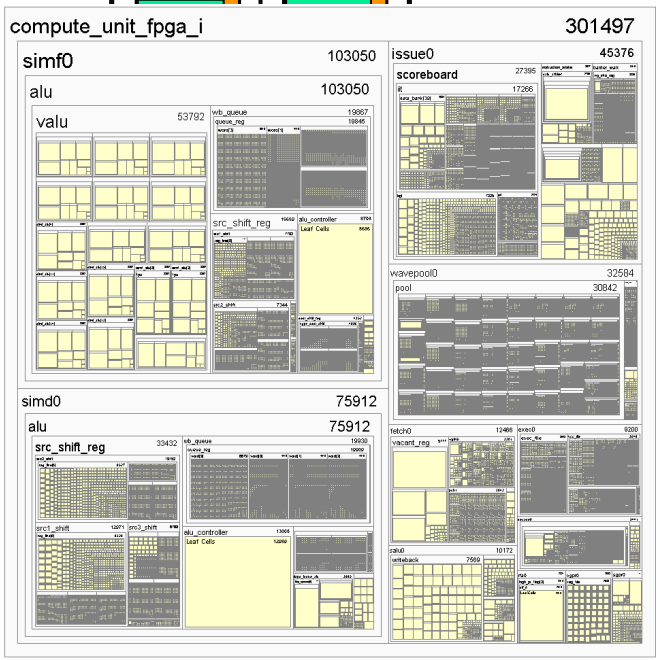
(b) Mapped to FPGA



Long running apps
S/W development
Prototyping

(c) Hybrid Design

Virtex-7 FPGA
1 CU @ 50 MHz
133K LUTs, 100K Registers



research

Legend

CU : Compute Unit; M0, M1, M2 : Memory Controllers

RTL | Hard Macro | BRAM | FPGA IP | C/C++ Models

Design Team

- Small initial design team (12 mo)
  - 5-person HDL team
  - 1-person software team
  - 1-person physical design team
- Added FPGA expert
- 3 undergrads extended the design
- Total duration: 36 months
- Area, Frequency, Performance, Power – NON GOALs

Software Compatibility

- Runs unmodified OpenCL programs
- All AMD APP SDK OpenCL benchmarks
- Many Rodinia benchmarks
- Easily extendable to add additional instructions

MIAOW Technical Overview

Demonstrate MIAOW is a credible GPGPU

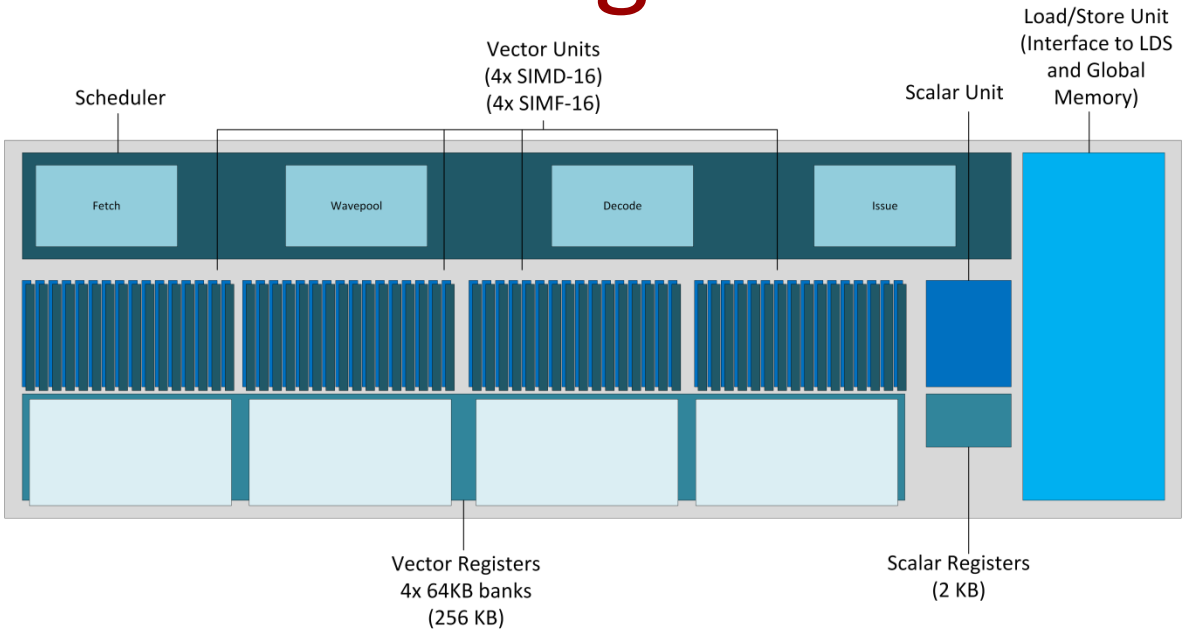
Implications and Possibility of Open Source Hardware

MIAOW vs. AMD Tahiti

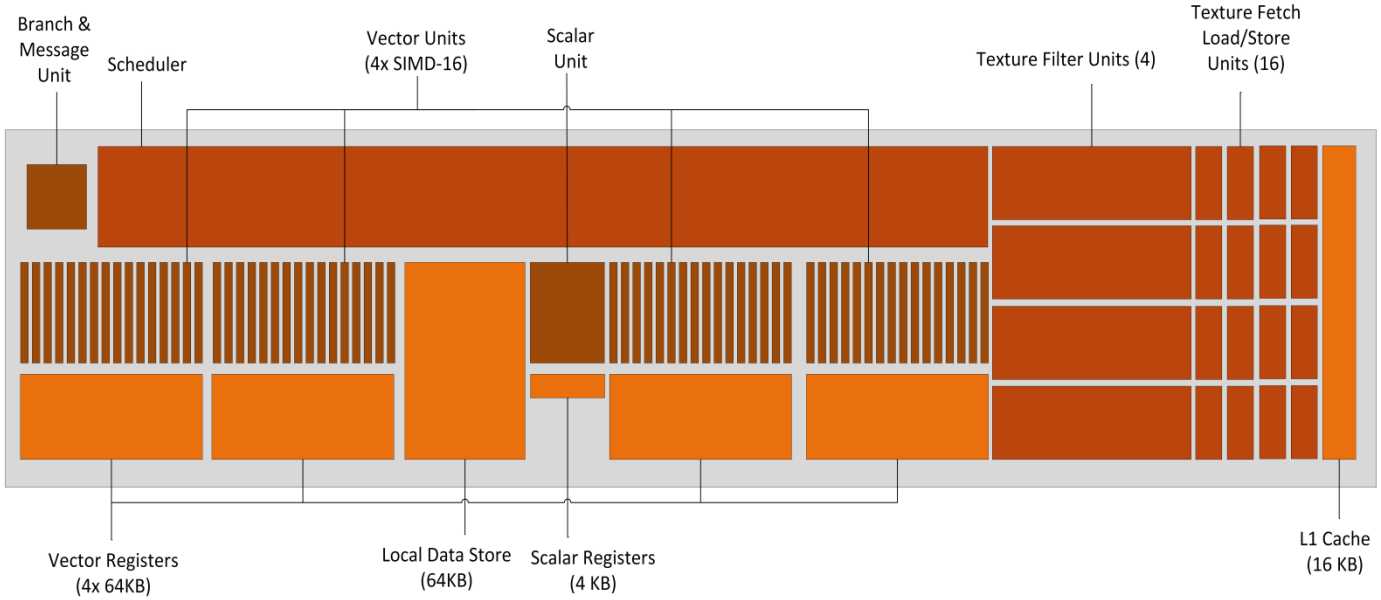
AMD's latest GPU CU architectures build upon Tahiti. Tahiti released in 2011.

CU Design

MIAOW

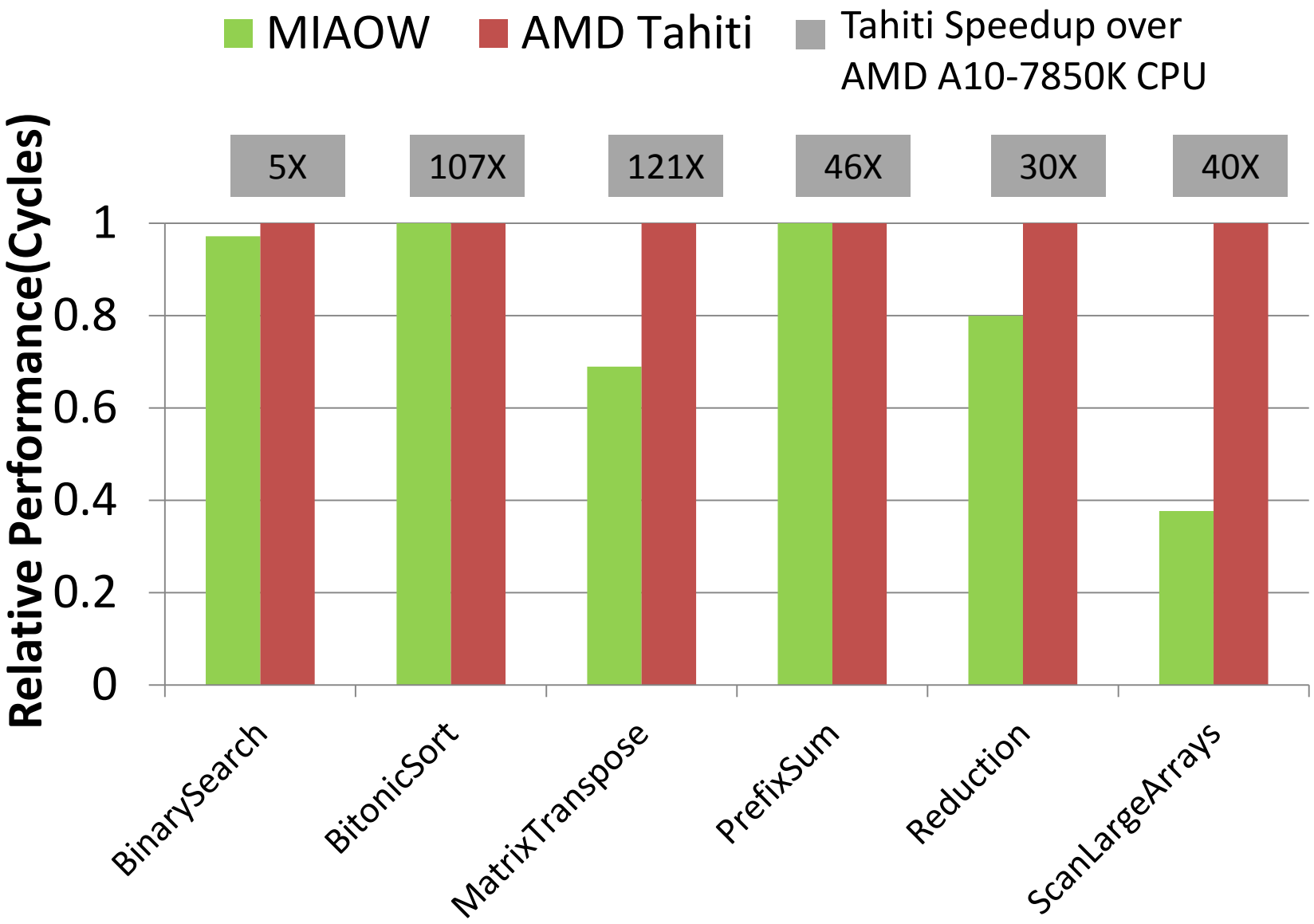


GCN CU*



**HOTCHIPS 2014*

Performance Comparison



Area Comparison

Tahiti CU area*: 5.02 mm^2 @ 28nm

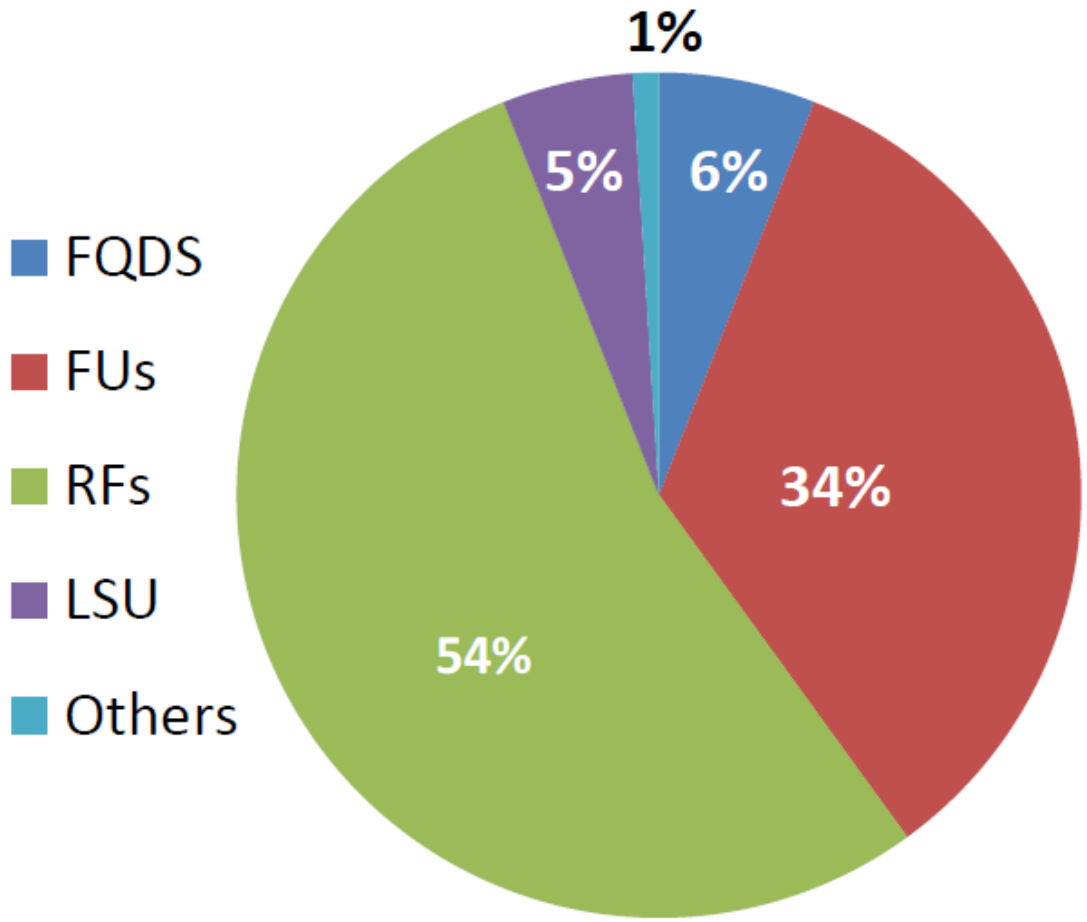
MIAOW CU area: 9.1 mm^2 @ 32nm

**Estimate from die-photo analysis and block diagrams from wccftech.com*

Area Comparison

Tahiti CU area*: 5.02 mm^2 @ 28nm

MIAOW CU area: 9.1 mm^2 @ 32nm



*Estimate from die-photo analysis and block diagrams from wccftech.com

Power

Tahiti CU power*: 0.52 W

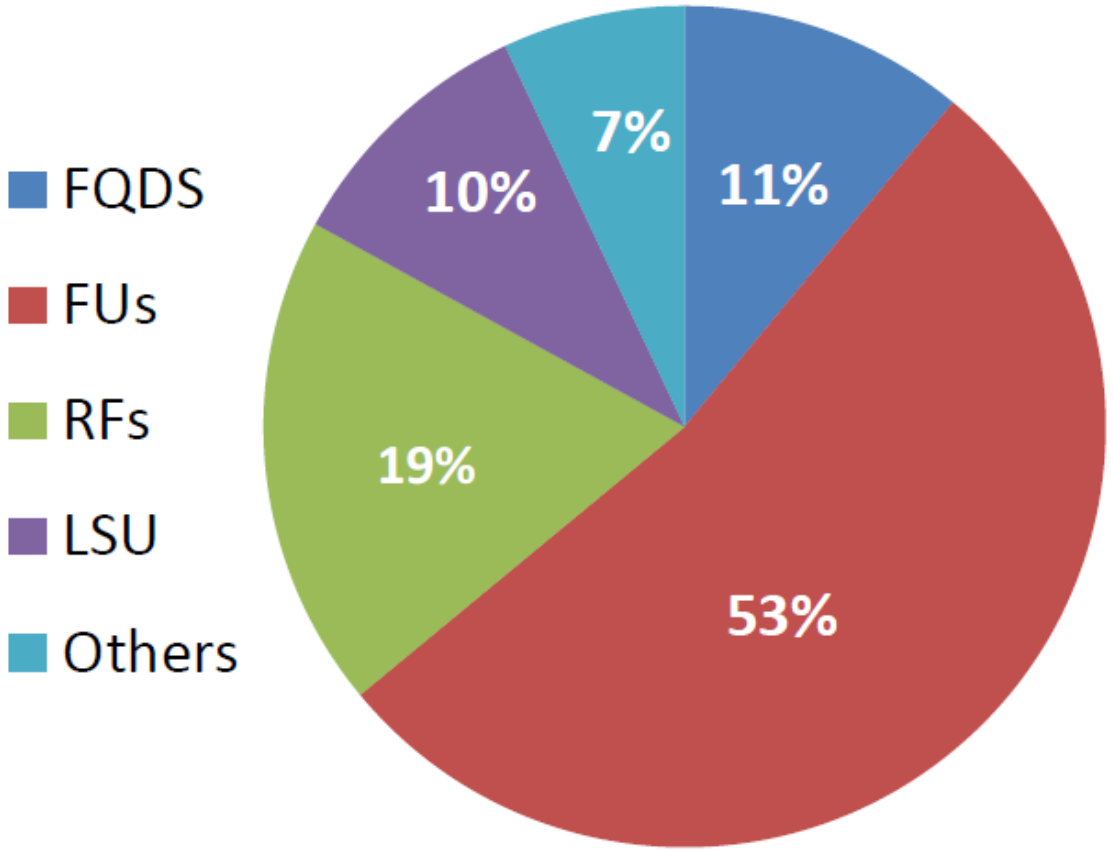
MIAOW CU Power: 1.1 W

** Ballpark estimate from TDP and occupancy*

Power

Tahiti CU power*: 0.52 W

MIAOW CU Power: 1.1 W



** Ballpark estimate from TDP and occupancy*

MIAOW is comparable to industry designs

MIAOW Technical Overview

Demonstrate MIAOW is a credible GPGPU

Implications and Possibility of Open Source Hardware

Lessons Learned

- It was surprising this was doable!
  - Microarchitecture design, HDL implementation, verification was not tedious
- Software toolchain being available was great
- We punted on physical design
- FPGA tools are still quite tedious to use

Implications for Industry

- Open Source Hardware GPU
  - Relevance to OpenCompute & Maker movement
- How can a HW startup benefit from MIAOW
  - Start with MIAOW and focus on innovative pieces from day one
- IP and Compiler
  - License under BSD, ISA is OK, compiler usable
  - How to avoid IP infringement?

Lessons from Open Source S/W

PHP, Linux, ruby, mysql, sqlite, apache, gcc
late 80s, early 90s

$0

Facebook, Twitter, Whatsapp, Instagram
Web 2.0

>$10 bill.

MIAOW, OpenCores, RISC-V etc..

$0

?

What drives Open Source Software?

Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects, Lakhani K and Wolf R. In Perspectives on Free and Open Source Software

- It's fun!
  - "**Enjoyment-based intrinsic motivation**, namely how creative a person feels when working on the project, is the strongest and most pervasive driver."
- It's valuable
  - "**user need, intellectual stimulation** derived from writing code, and improving programming skills are top motivators"

Conclusion

- MIAOW is transformative for GPU research
- Its role in open source hardware movement?
- Are open source hardware chips feasible?
- More community support → First Open Source Silicon GPU Chip

www.miaowgpu.org

Journal article *(ACM TACO 2015)*: Enabling GPGPU Low-Level Hardware Explorations with MIAOW: An Open-Source RTL Implementation of a GPGPU

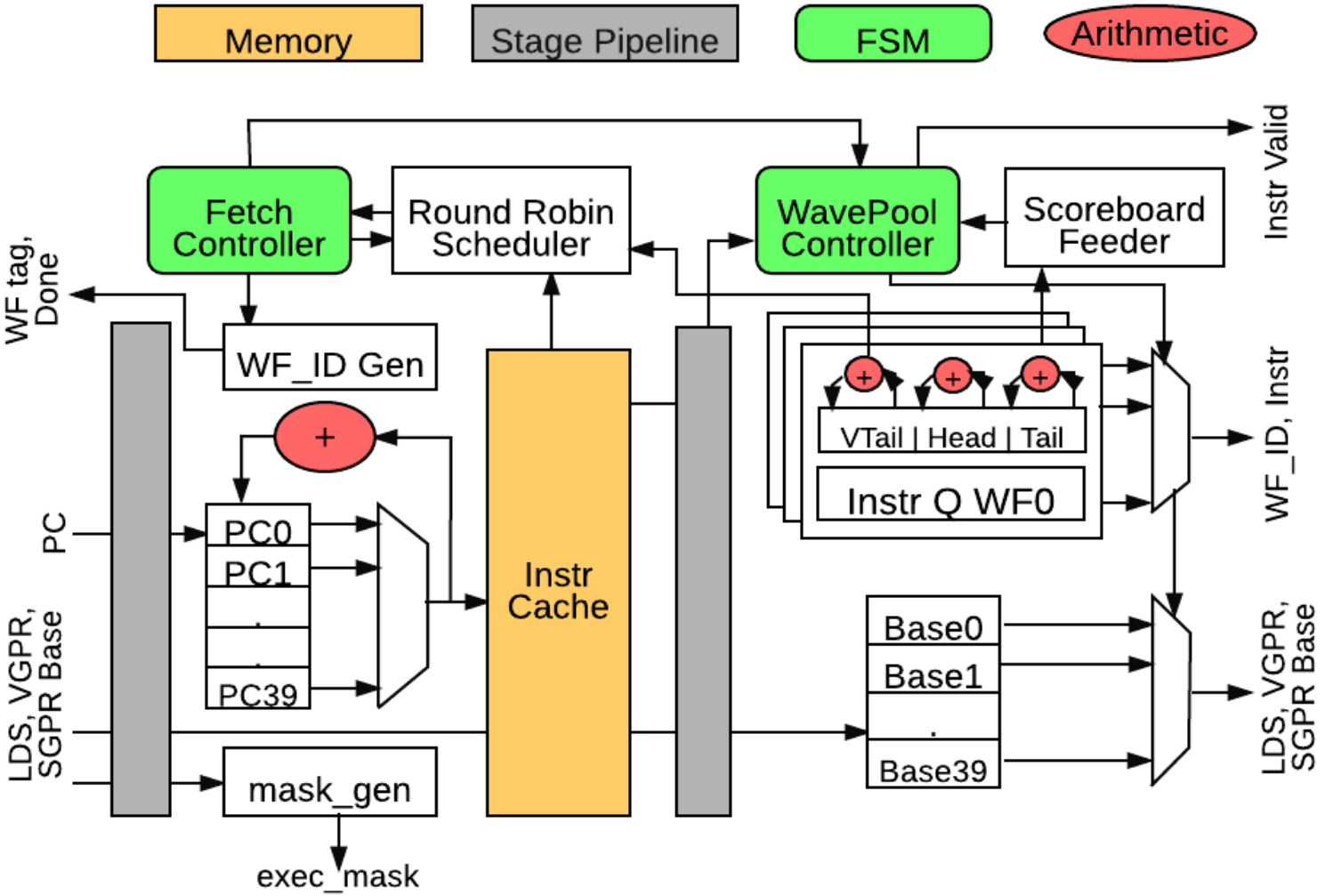
3.9 FPS on FPGA @ 50 MHz

23 FPS in simulation @ 222 MHz

MIAOW

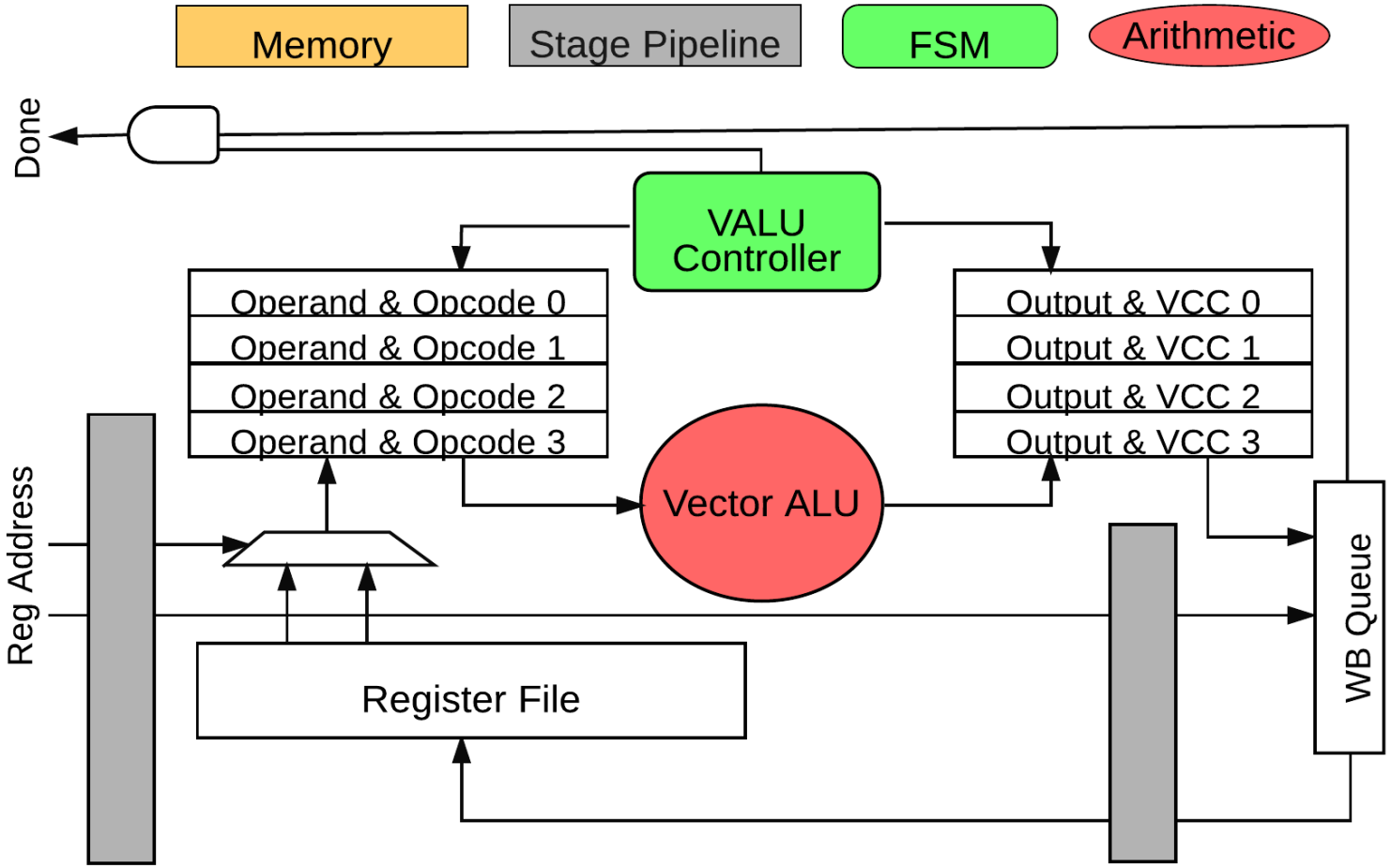
Back Up Slides

Fetch & Wavepool



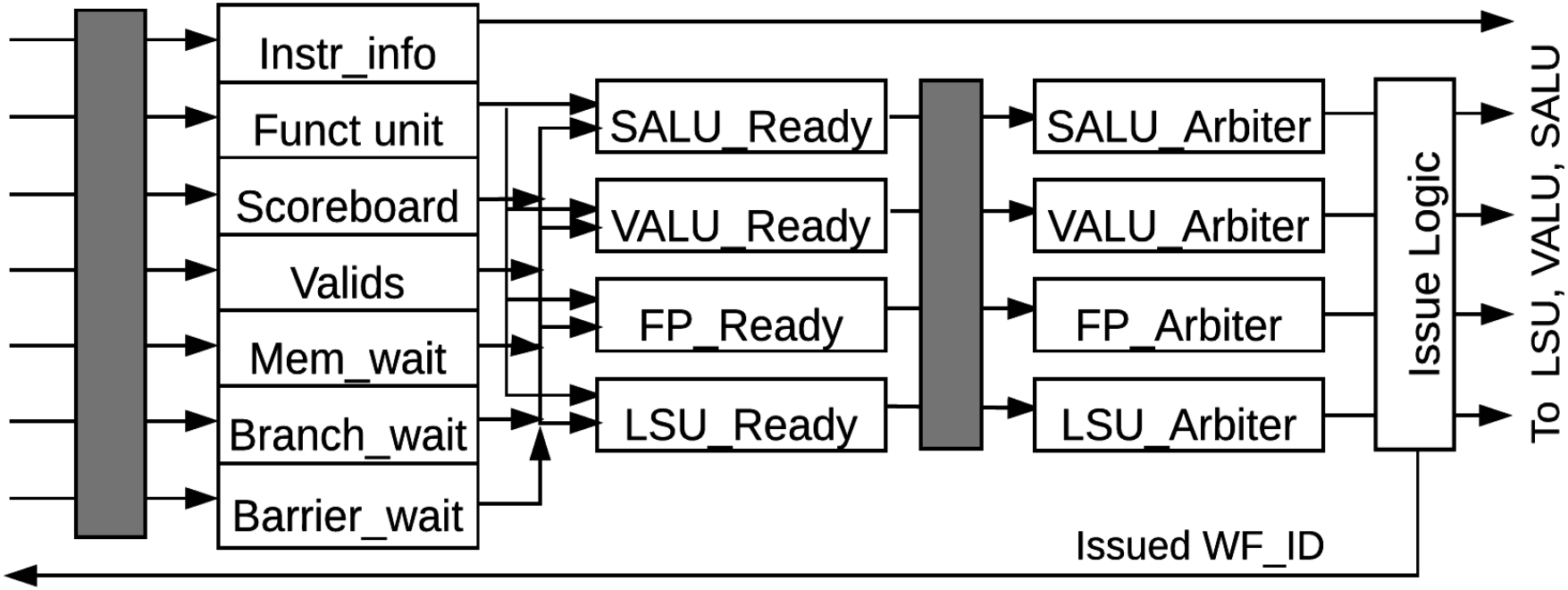
Fetch & Wavepool

Vector ALU/FPU



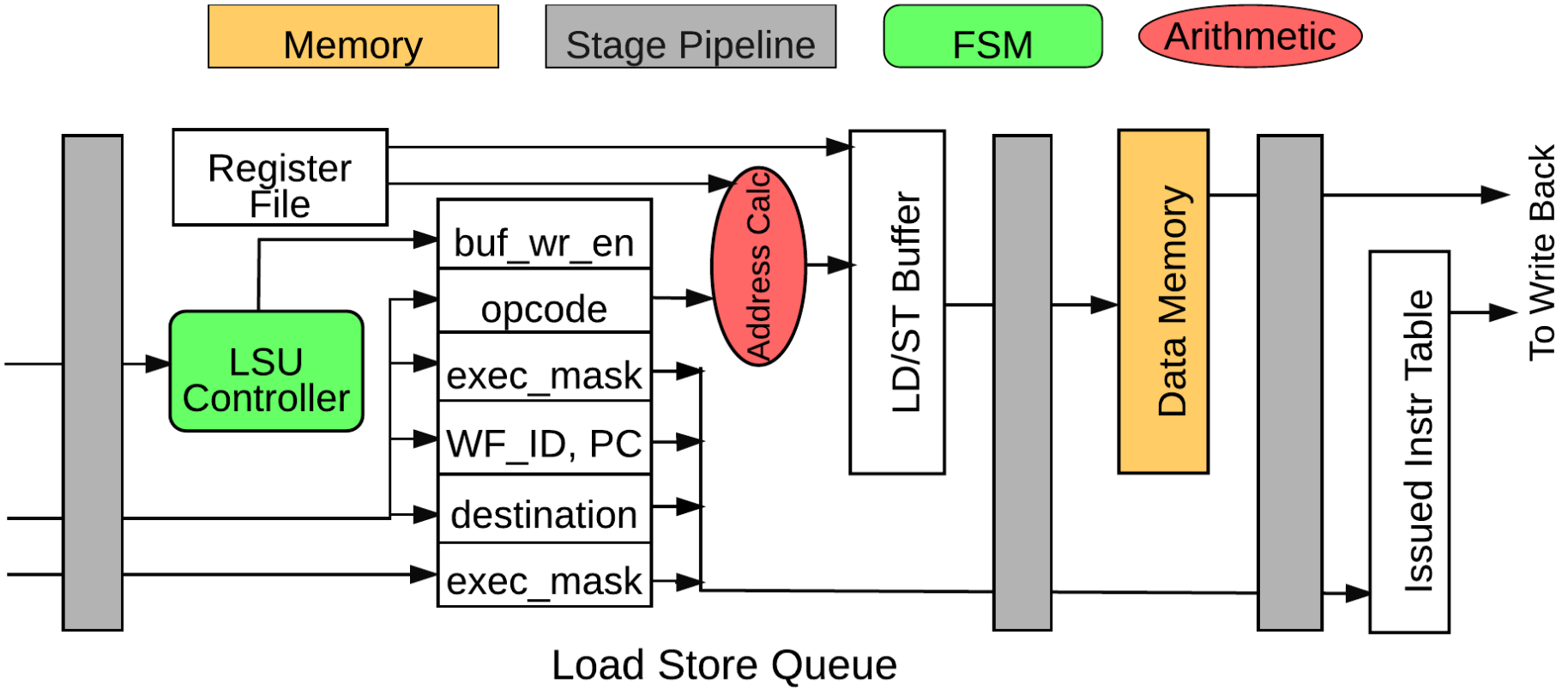
Vector ALU/Vector FPU

Issue



Issue: Wavefront Scheduling and Arbiter

Load/Store Unit



www.miaowgpu.org

HOTCHIPS 2015

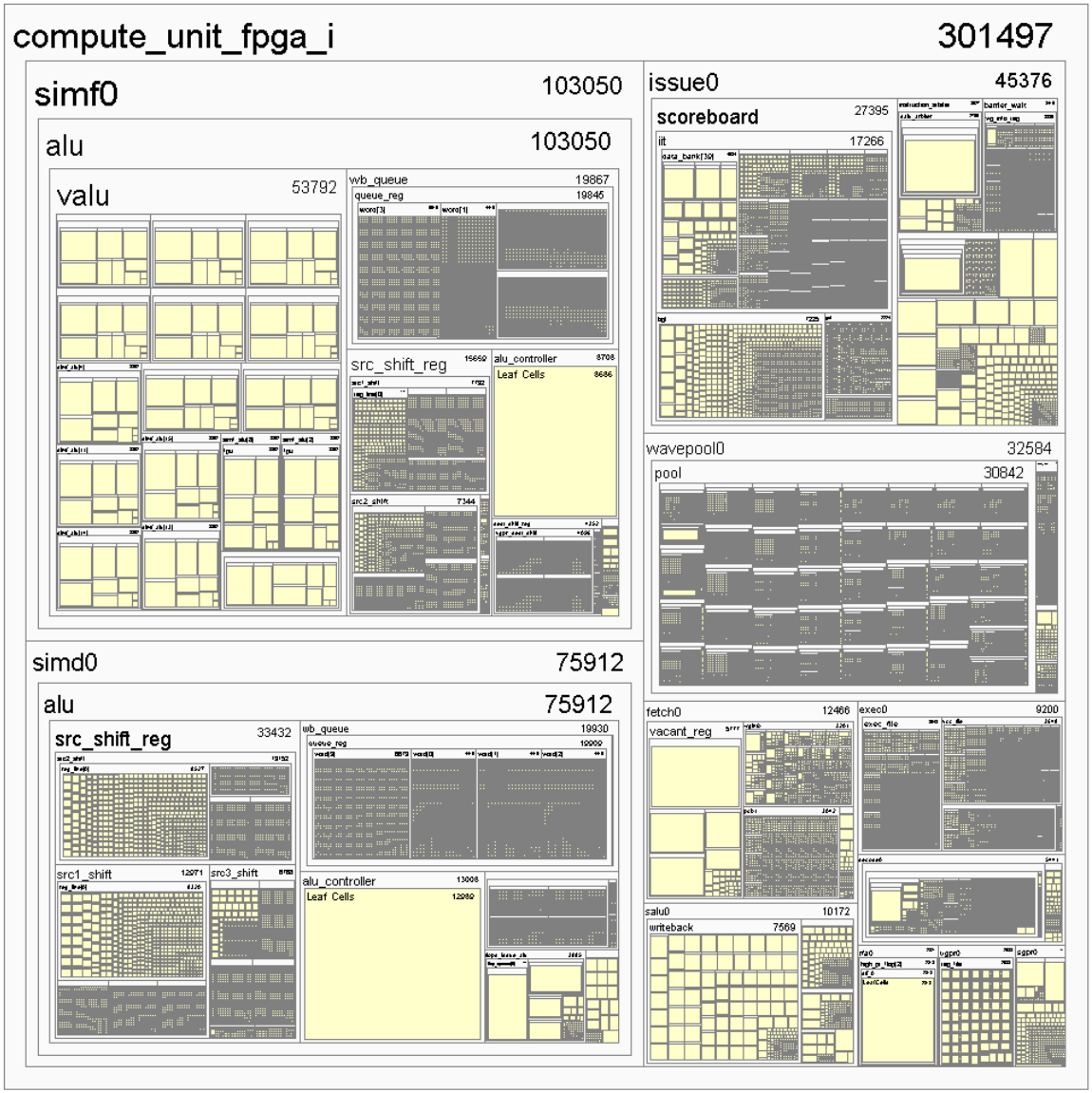
Many technical details in this publication: TACO 2015: Enabling GPGPU Low-Level Hardware Explorations with MIAOW: An Open-Source RTL Implementation of a GPGPU

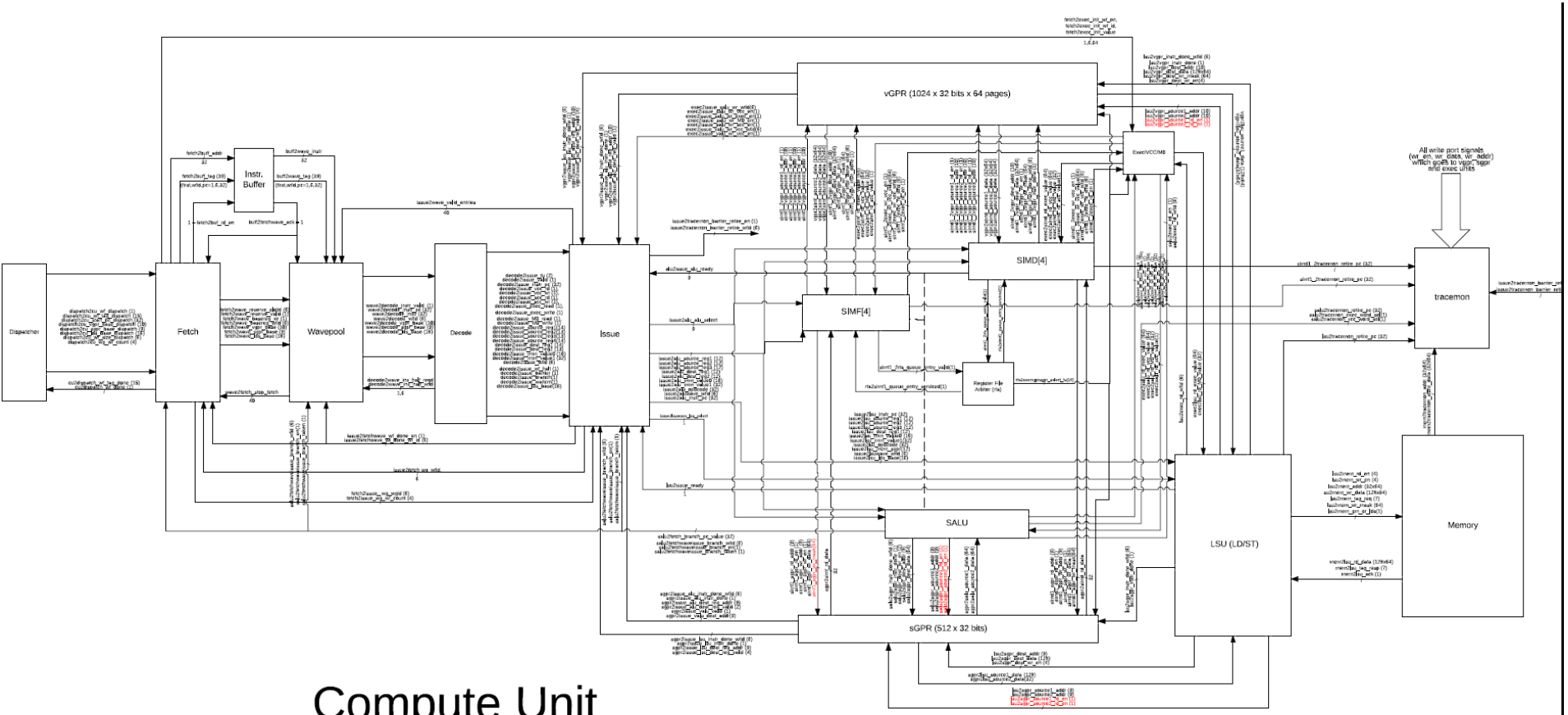
Virtex7-based FPGA – Neko

- 1 CU design
- 50 MHz
- Utilization:

  LUTs 133K

  Registers: 100K





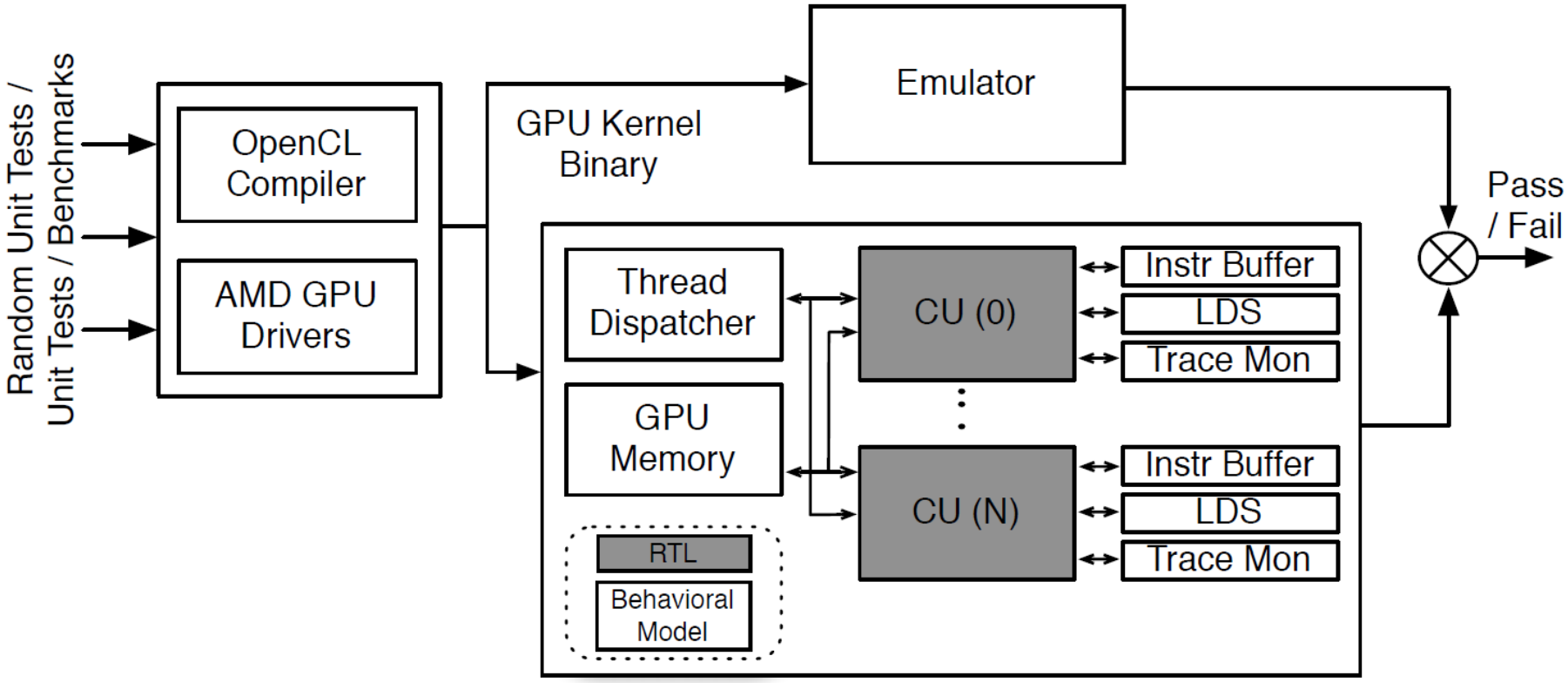
Compute Unit

Flexibility

Design choice	Realistic	Flexibility	Area/Power impact
Fetch bandwidth (1)	Balanced†	Easy to change	Low
Wavepool slots (6)	Balanced†	Parametrized	Low
Issue bandwidth (1)	Balanced†	Hard to change	Medium
# int FU (4)	Realistic	Easy to change	High
# FP FU (4)	Realistic	Easy to change	High
Writeback queue (1)	Simplified	Parametrized	Low
RF ports (5,4)	Simplified	Hard to change	High
RF ports (SRAM) (1)	Realistic	Hard to change	Low
Types of FU	Simplified	Easy to change	High

†*Fetch optimized for cache-hit, rest sized for balanced machine.*
Numbers in parenthesis indicate the design parameters.

Verification



As a Research Tool

Direction	Research Idea	MIAOW enabled findings
Traditional µarch	Thread-block compaction (TBC)	• Implemented TBC in RTL • Significant design complexity • Increase in Critical Path length
New Directions	Circuit-Failure Prediction (Aged SDMR)	• Implemented entirely in µarch • Works elegantly in GPUs • Small area, power overheads
	Timing Speculation (TS)	• Quantifies error-rate on GPU • TS framework for future studies
Validation of Simulator studies	Transient Fault Injection	• RTL Level Fault Injection • More Gray area than CPUs • Silent data corruption seen

Virtex7-based FPGA – Neko

Virtex7 FPGA

1 CU @ 50 MHz

133K LUTs, 100K Registers